

System And Method For Implementing A Natural Language User Interface

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. Patent Application S/N
5 09/880,216 filed on June 13, 2001, which issued as U.S. Patent No. 6,691,111 on
February 10, 2004. U.S. Patent No. 6,691,111 claims priority from and is related
to the following prior application: Natural Language User Interface, United States
Provisional Application No. 60/215,222, filed on June 30, 2000. These prior
applications, including the entirety of the written descriptions and drawing
10 figures, are hereby incorporated into the present application by reference.

BACKGROUND

1. Field of the Invention

This invention generally relates to user interfaces. More particularly, the
15 invention provides a system and method for accessing software applications with
limited user input. The invention is particularly well-suited for use in Personal
Digital Assistants, mobile communication devices, cellular phones, and wireless
two-way e-mail communication devices (collectively referred to hereinafter as
"hand-held devices"). The invention provides utility, however, in any device
20 which requires a user interface.

2. Description of the Related Art

In traditional hand-held devices, a typical user interface is graphical in nature. An example of a known graphical user interface 10 is shown in FIG. 1. The known graphical user interface 10 requires a user to scroll through a ribbon of icons, and select the desired application by "clicking" on the icon associated with it. Once in an application, the user cannot open a second application without (a) exiting the application, (b) returning to the graphical user interface 10, and (c) selecting an icon associated with the second application.

10

SUMMARY

A system and method are provided for implementing a natural language user interface. The system includes a plurality of software applications, a means for entering text, and a natural language search engine. The natural language search engine is configured to receive an abbreviated textual command from the text entering means, match the abbreviated textual command with a desired application command, and launch one of the software applications by executing the desired application command.

15

BRIEF DESCRIPTION OF THE DRAWINGS

20

FIG. 1 is a known graphical user interface;

FIG. 2 is a block diagram showing exemplary components of a natural language user interface system according to the present invention;

FIG. 3 shows an exemplary home screen, including an exemplary graphical dialog box;

5 FIG. 4 shows an exemplary two-part abbreviated textual command;

FIG. 5 is a table illustrating exemplary application commands and exemplary associated application tags;

FIG. 6 is a flow diagram showing an exemplary method of matching an abbreviated textual command with a desired application command and application
10 tag using a probability analysis;

FIG. 7 shows an exemplary data set stored in a natural language database;
and

FIG. 8 is a flow diagram representing an exemplary user interface having both an icon ribbon and a natural language interface.

15

DETAILED DESCRIPTION

Referring now to the figures, FIG. 2 is a block diagram showing exemplary components of a natural language user interface system 30 according to the present invention. This system 30 may include a natural language search
20 engine 32, an email compose application 34, an address book application 36, a

calendar application 38, a home screen 40, and a natural language database 42. The natural language search engine 32 is preferably a software module that interacts with the other software applications and databases in the system. Operationally, the natural language user interface system 30 connects all of the software applications 34, 36, 38 via the natural language database 42 and the natural language search engine 32. Preferably, the natural language search engine 32 may be accessed by a user from the home screen 40 or from within any software application 34, 36, and 38. In one possible embodiment, the user accesses the natural language search engine 32 by entering abbreviated textual commands in a graphical dialog box 44 that is available on the home screen 40 and is accessible in the software applications 34, 36, and 38. The text may be entered into the graphical dialog box 44 using a keypad, an auxiliary input device such as a thumbwheel, or any other known input device.

FIG. 3 shows an exemplary home screen 40, including a graphical dialog box 44. Using the graphical dialog box 44, software applications 34, 36, and 38 may be quickly launched from any point within the system by entering a textual command, preferably in the form of an abbreviated textual command. For example, a user may wish to open the calendar application 38 while he or she is using the email compose application 34. To execute this operation, the user could access the graphical dialog box 44, and enter an abbreviated textual command associated with the calendar application, such as "c" for calendar or "app" for

appointment. Referring again to FIG. 2, the natural language search engine 32 then compares the abbreviated textual command entered in the graphical dialog box 44 with a command list stored in the natural language database 42, and either executes the associated operation or presents the user with a list of possible commands.

FIG. 4 shows an exemplary two-part abbreviated textual command 50. The user enters the exemplary two-part abbreviated textual command 50 in the form of "x_y" (where "_" represents a space). The natural language search engine 32 relates the first input component 52 ("x") to a possible application command 56, and the second input component 54 ("y") to a possible application tag 58. An application command 56 refers to a discrete software application or sub-application in the system 30, such as the email compose application 34, address book application 36, or calendar application 38. An application tag 58 refers to data associated with the discrete software application or sub-application; for example, retrieving an individual's name stored in the address book application 36.

FIG. 5 is a table 60 illustrating exemplary application commands 56 and exemplary associated application tags 58. For example, possible application tags associated with the command "compose" could be names or email addresses that are stored in a database accessed by the email compose application 34.

FIG. 6 is a flow diagram 70 showing an exemplary method of matching an abbreviated textual command with a desired application command and application tag using a probability analysis. Using this method, the natural language search engine 32 interprets which application command 56 and application tag 58 (collectively referred to hereinafter as "user command") is likely desired by the user based on a historical probability. Where a user command has a high likelihood, the natural language search engine 32 will invoke that user command rather than expending processor resources searching for other alternatives.

In step 72, the user enters an input, preferably in the form of "x_y" as described above with respect to FIG. 4. In steps 74 and 76, the natural language search engine 32 attempts to associate the input with an exact match in a list of user commands stored in the natural language database 42.

FIG. 7 shows a exemplary data set 100 stored in the natural language database 42. This data set 100 includes a user input record 102, a user command record 104, and a probability factor record 106. The user input record 102 is a historical record of all of the abbreviated textual commands previously entered by the user. The user command record 104 includes a set of user commands associated with each abbreviated textual command, and the probability factor record 106 includes a probability factor associated with each user command. For example, the abbreviated textual command "e_j" has two associated user commands, "email john" and "email jane." The user command "email john" has a

probability factor of 0.8, and the user command "email jane" has a probability factor of 0.2.

The probability factors are assigned by the natural language search engine 32 based on the number of times a user executes a given user command after entering a given abbreviated textual command, and preferably the amount of time since the user command was last invoked. If, after a set number of entries, a user command has always been associated with the same user input, the user command has a probability factor of 1.0 and is an exact match. For example, if a user always enters "e_j" to execute the user command "email john," after a set number of entries this pair will be assigned a probability factor of 1.0 and will be considered an exact match.

Referring again to FIG. 6, an exact match in step 76 causes the natural language search engine 32 to execute the user command automatically at step 80 after setting the exact match to the user command at step 78. If the natural language search engine 32 finds more than one or no matches at step 76, however, then a probability analysis at steps 82 and 84 is undertaken to find a suitable match. The purpose of the probability analysis is to narrow the search results of step 76 to the user command most probably desired by the user. As a user configurable default, if any one of the results of step 76 has more than a 0.5 probability factor, then that result is a probabilistic match. The probabilistic match is then set to the user command at step 78, and executed in step 80.

If no probabilistic match can be determined in steps 82 and 84, however, then the natural language search engine 32 will display a list of possible commands to the user in step 86. In step 88, the user may select from the list of possible commands, or alternatively may narrow the list by entering more text. In
5 a preferred embodiment, the list of possible commands displayed in step 86 is presented in a manner that allows the user to quickly select one of the choices. One such method is to present the command list to the user arranged according to the frequency of use as determined by the probability factor. In the event that a user enters text that does not match any user commands in the natural language
10 database 42, the natural language search engine 32 will preferably present a list of generic commands that it is capable of executing. Alternatively, at step 86 the user is permitted to escape and return to step 72 to enter a different input by executing an escape sequence 90.

After a user command has been executed in step 80, the user may return to
15 step 72 by pressing an escape key, or by some other suitable escape sequence 90. At step 92 the probability factor record 106 is adjusted appropriately to reflect the users actions.

In an alternative embodiment, the method for matching an abbreviated textual command to a desired user command utilizes set abbreviations assigned by
20 the user based on his or her preferences. In this embodiment, the user may, at his or her option, associate abbreviations with user commands, preferably as part of a

setup procedure for the natural language database or as a user option. For example, the user may chose "e" to always represent a command to launch the email composer application 34, or possibly "e_j" to always represent the user command "email jim." In this manner, only two keystrokes are required to invoke
5 the email composer application 34, and select the addressee "jim." In the event that a user enters text that does not match any system commands or abbreviated commands set by the user, the natural language search engine 32 may return an error message and prompt the user to enter the input again, or show a list of possible command choices. The list of possible command choices may include
10 the most recently entered user commands or a list of the generic commands that the natural language search engine 32 is capable of executing. Upon selecting a user command, the user will preferably have the option of assigning his textual input as a new abbreviated command.

In a further embodiment of the present invention, a list of frequently used
15 commands is displayed by the natural language search engine 32 as soon as the user begins entering text in the graphical dialog box 42. The user then has the option to continue typing or choose from the list of commands.

In another embodiment of the present invention, the user is able to create short cut keys associated with a desired user command. In this embodiment, the
20 natural language search engine 32 enables a user to associate key strokes with a user command within the natural language database 42.

FIG. 8 is a flow diagram 110 representing an exemplary user interface having both an icon ribbon and a natural language interface. In this embodiment, the home screen includes an icon ribbon, such as the icon ribbon in the known graphical interface 10 shown in FIG. 1. While in the home screen at step 112, the user has the option of selecting an icon in step 114, or activating the natural language search engine 32 by entering text in step 116. Preferably, a graphical dialog box 44 is opened by the natural language search engine 32 as soon as the user begins entering text. Alternatively, the natural language search engine 32 may be activated by some other means; for example by selecting a hot key or designated key combination, or, possibly, by selecting a designated icon.

In this embodiment, an application may be executed in step 118 by either the icon ribbon or the natural language search engine 32. Once an application is launched, the user has the option in step 120 to launch a different application as described above with respect to FIGs. 2 and 3. For example, a user may launch the email compose application 34 in step 114 by either selecting the appropriate icon or entering a command using the natural search engine 32. Once the email compose application 34 has been executed in step 118, the user may then wish to launch the address book application 36 without first exiting back to the home screen. In step 120, the user has the option of launching the address book application 36 by activating the natural language search engine 32 from within the email compose application 34.

The invention has been described with reference to the preferred embodiments. Those skilled in the art will consider improvements, changes and modifications in view of the foregoing description. Such improvements, changes and modifications are intended to be within the scope of the claims.